



کانال امیرکبیر جزوه



@AmirkabirJ

@Amirkabiriha

دسترسی رایگان به فیلم‌های آموزشی تدریس جزوه:



www.polycoursium.com

- $\downarrow$ I/O - controller, CPU

نسخه انفرادی
کاربرد در معنی یادگار

بسم الله الرحمن الرحيم

OS (سیستم عامل) سے نرم افزار سمیٹے سے واسطے میں سخت اثر دار انسان

دوربین منبع → متن CPU, Ram, دستگاه I/O

online spooling & offline spooling :-

الغالبه، نوار، عرق انتقال کار - استقامت از دین

زمان روشن کار، - " ↓

انداع سراسر آنها از نظر منابع معدنی

cpu bounded ← CPU پر بار (کام یا کڑی کام)

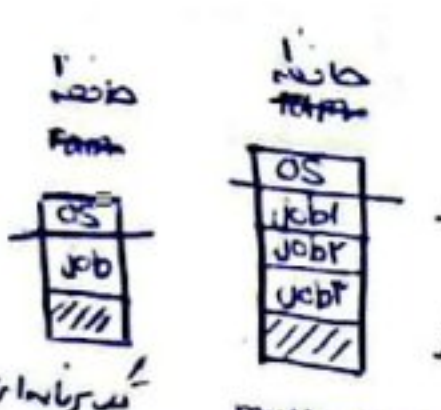
(normalizing) $I/O \leftarrow I/O$ bounded

* انگریزی کا ایسی - صرف multi program

cpu bounded, لا يتوقف على الذاكرة

5. L^1 norm bounded

البيانات: CPU و I/O و البيانات



→ یقیناً زمان
صرفہ ہے اس
اصل اصل تا آخر
ایک بار ۲۷
گرامر

جمع اوصی کا حصہ محفوظ توبہ PC ایزان تحت
ذخیرہ کا حصہ دوسرا، مخصوص
انتقال نواریہ توبہ ایم ایئر ب PC اصلی
چھ لائزنگ کا حصہ، ترتیب توبہ PC اصلی دیگر
ذخیرہ ضمیمہ توبہ PC اصلی دوسرا
انتقال نواریہ توبہ ایم ایئر ب PC ایزان
چھ ضمیمہ توبہ PC ایزان

(batch system) مربوط به بلادرز

↑ میرا اسباق رتہ pc صاف ہے اور زبان

چهارم: روش در PC (پایان)

محيطات /

۱) این قولها، تأدیفات نیستند (تأدیفات: تادیب، تنبیذ)

(بدریل) کوئی ضابطہ بصورت رسم الہیہ (بسم)

اولویٰ بند ۴ X (مذلل القواعد از نوار)

offline
Spooling
↓
online

۱- اطلاعات برنامه در آن حافظه
 نوع خاص از سیستم حافظه است
 ۲- چه برنامه به طور موقت در حافظه
 ۳- انتقالی که بعد از اتمام CPU

(multi programming) $\rightarrow$ running OS $\rightarrow$

← os استعمال کرتا ہے (time sharing)

← " چند صفت های جدیدی

ing/threading)

۵. چند بنای / پر دانه

• (Peak time) $\frac{1}{\sqrt{2}}$ " "

(networking) &

(distributed) توزیع شده

١٠٠٠

توابع سابق درجه

1. $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

عَنْ أَحْمَدَ بْنِ حَنْبَلٍ

۱- افزایش سن

من کتب و اسناد خطی

۱- یحییٰ بن علی

```

graph TD
    subgraph Users
        direction LR
        U1[user]
        U2[user]
        U3[user]
    end
    MW[middle ware]
    U1 --> MW
    U2 --> MW
    U3 --> MW
  
```

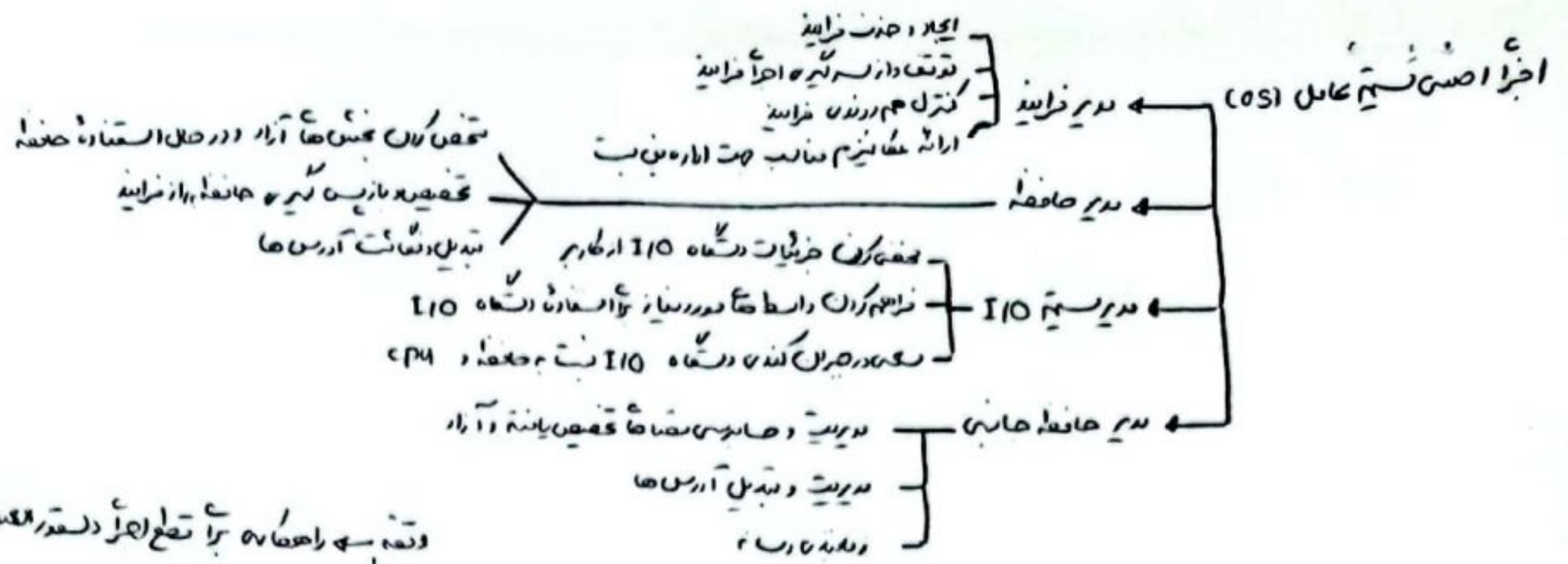
0.5	0.5	0.5
5	5	5

○ اُنس و انس کے مابین (میں نے) ذکر کیا ہے کہ

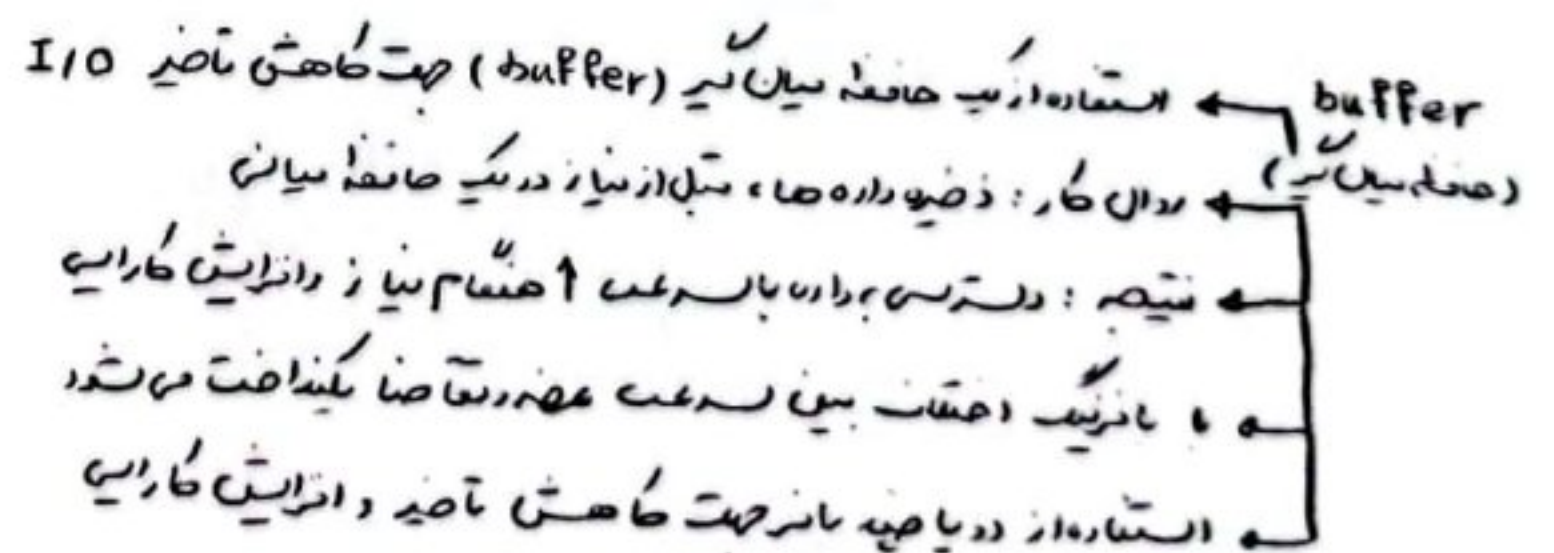
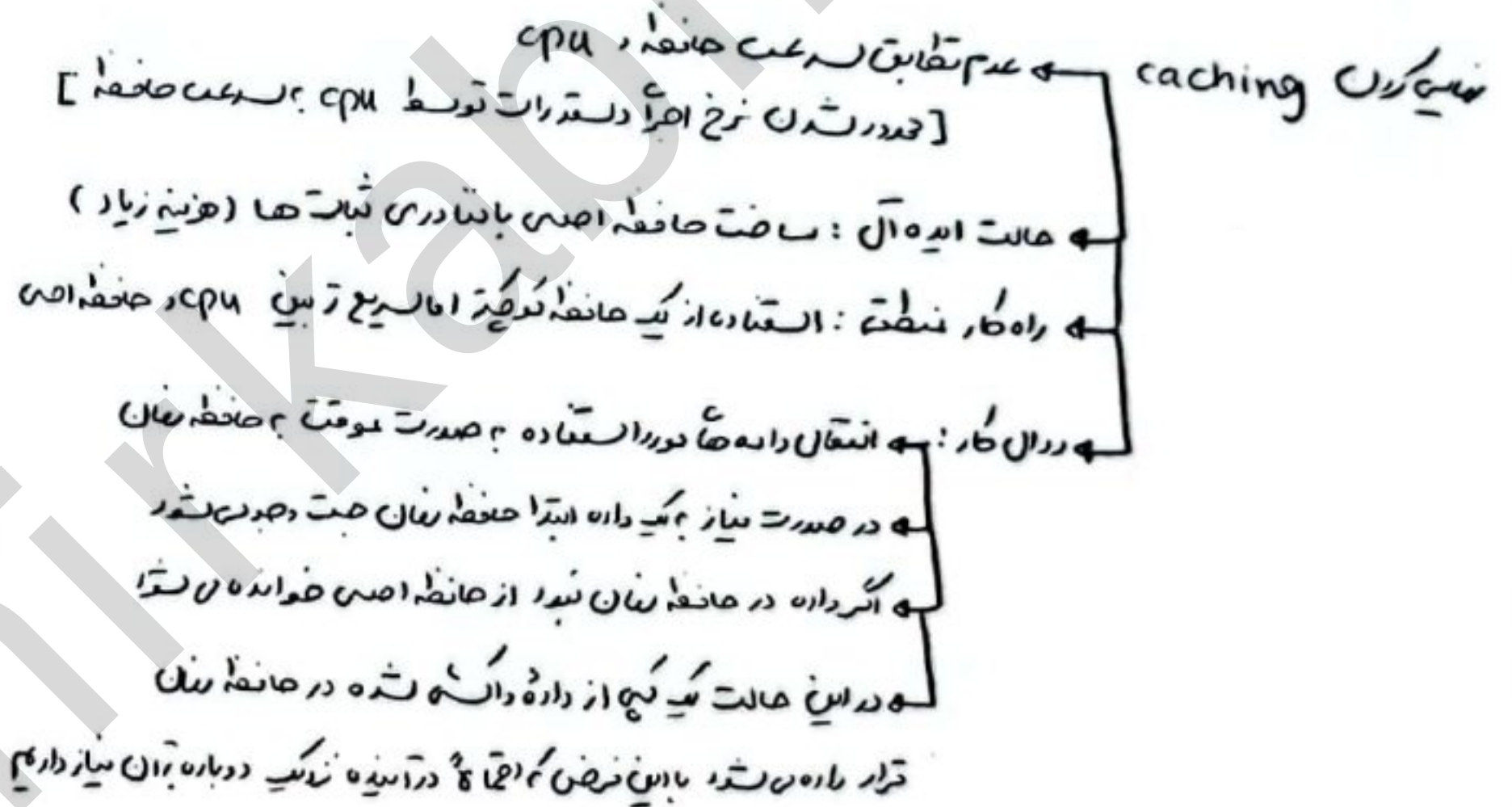
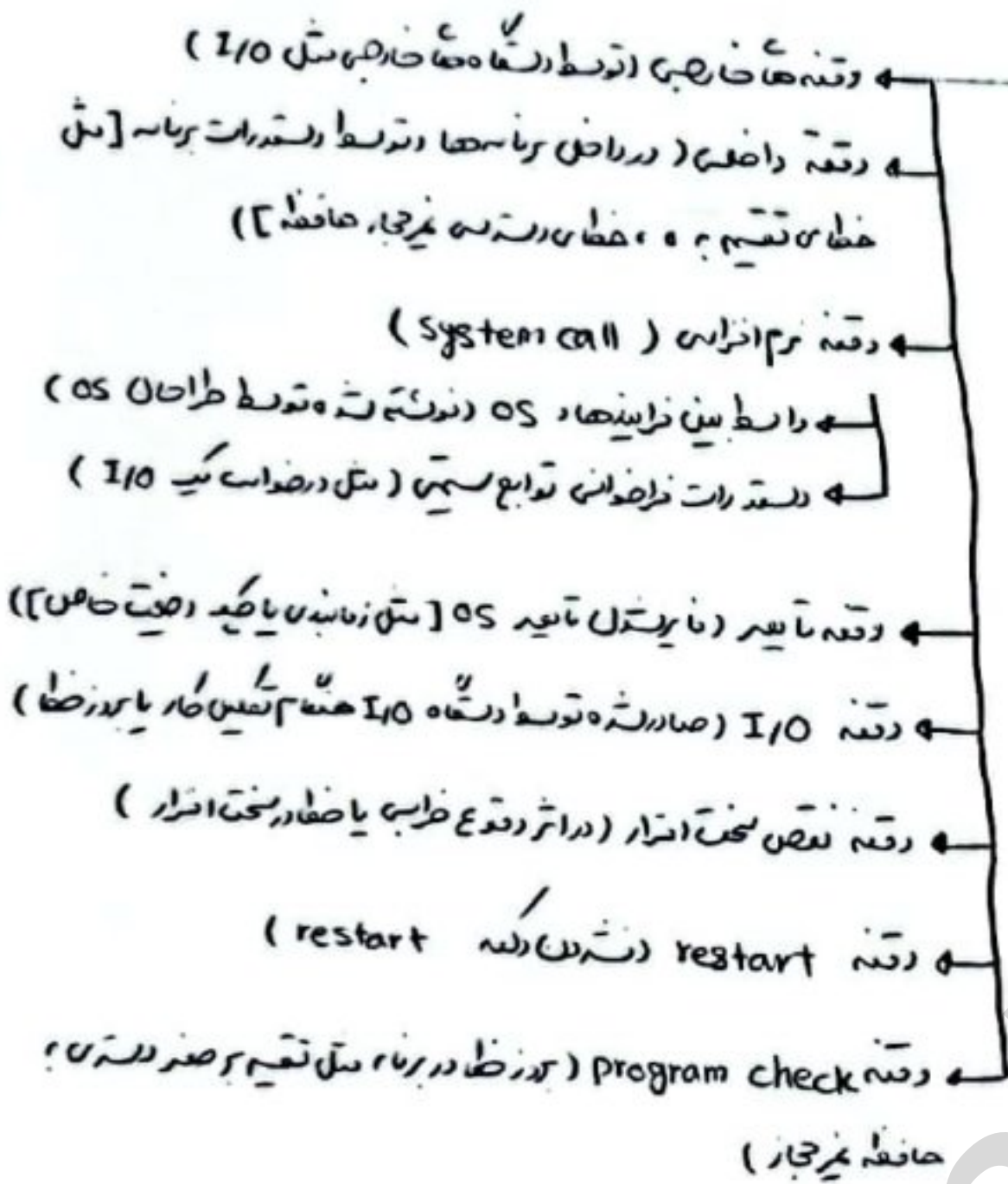
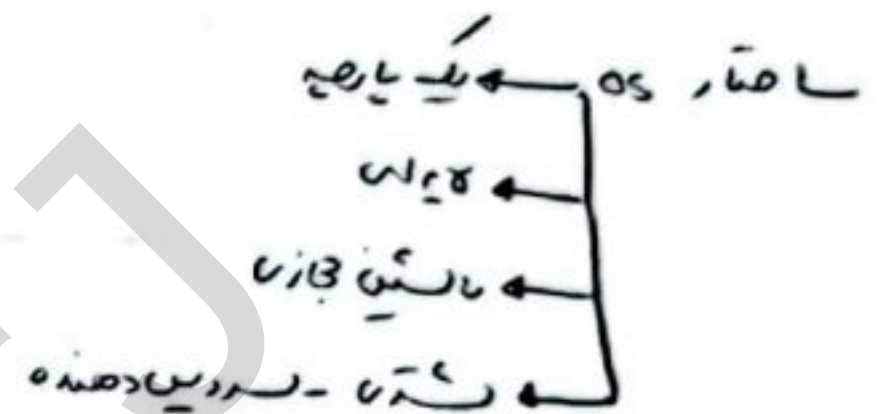
• دتہ ۵۵ ادریں بیغ (حافظہ) رابہش دار و متغیر

اور نواب لہرانہ سے تعلق

user	user	user
middle ware		
0.5	0.5	0.5
5	5	4



دقت: راهکارها برای رفع ایراد عملکرد عمل‌ها توسط CPU برای اجرا و توقف سرویس‌ها می‌باشد



فرایند از لحاظ اینکه کار توسط فرایند انتخاب دارد برودن مراحل مختلف در شود تا محاسبه اجرا آن به صورت کامل حالت سر یابد

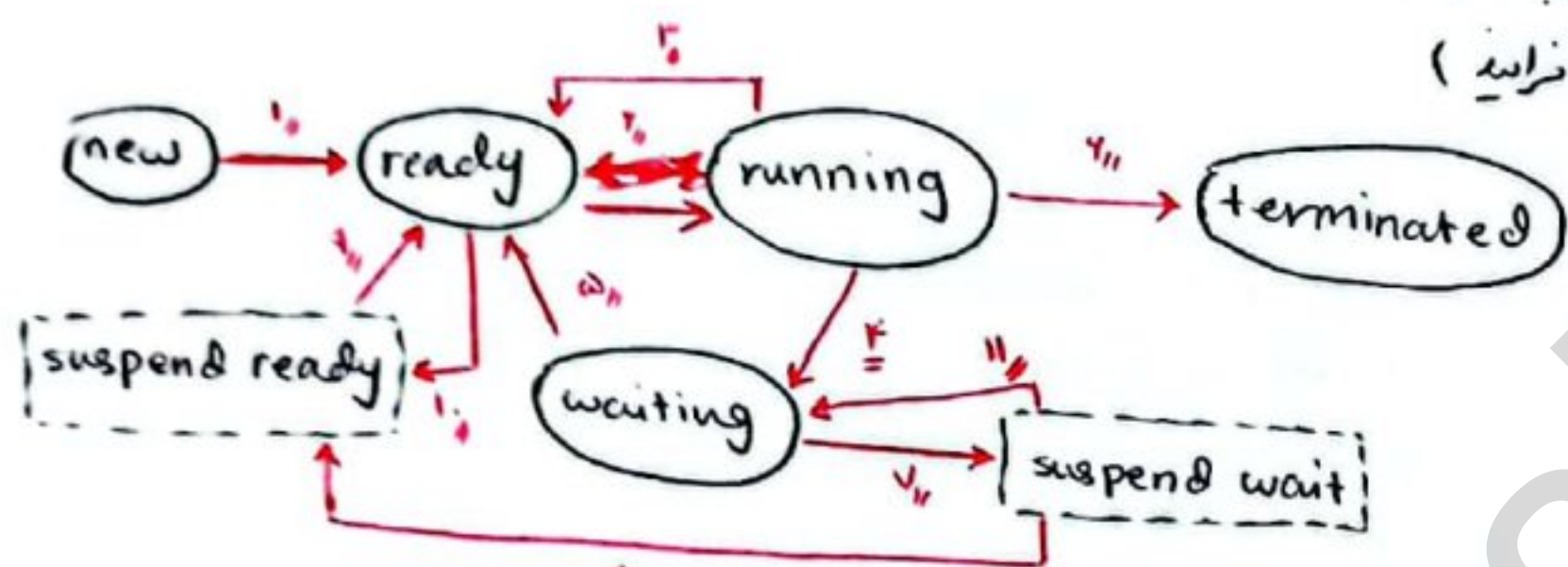
P.T به درج اطلاعات مربوط به فرایند توسط OS

اختصاص یک دارم در Process Table هر فرایند با عنوان

Process Control Block

- شماره فرایند
- اولویت فرایند
- ثابت های برنامه
- وضعیت فرایند
- اطلاعات منابع اختصاصی
- آدرس های خاص برنامه
- داده های
- اطلاعات مربوط به مدیریت حافظه

- 1 new (فرایند در حال ایجاد شدن)
- 2 Ready (فرایند درون حافظه، در اختیار داشتن منابع، منتظر CPU)
- 3 Running (در اختیار داشتن CPU و در حال اجرا)
- 4 blocked یا waiting (منتظر وقوع یک رخداد [مثلاً اختصاص منبع، یا تکمیل I/O])
- 5 terminated (انجام امر فرایند)



suspend برای فرایند ها که بیرون از جدول فرایند در حافظه می باشد

- 1 سیستم ماژول در حافظه یک فرایند می برد
- 2 CPU توسط برنامه ریز (job scheduler)
- 3 اختصاص CPU به فرایند
- 4 CPU توسط برنامه ریز (process scheduler)
- 5 یک مرتبه CPU از فرایند
- 6 فرایند اختصاصی CPU را بر می گرداند
- 7 منتظر وقوع رخداد فرایند می ماند
- 8 فرایند با اولویت بالا از راه برگرد

- 4 suspend wait (فرایند در حالت انتظار، از حافظه به یک منتقل شد)
- 5 suspend ready (فرایند به علت کمبود حافظه منتقم از حالت اجرا به ready و این حالت در دور فرایند می رود و یک در حالت منتظر قرار دارد، حالتی در در نظر گرفتن رخداد)

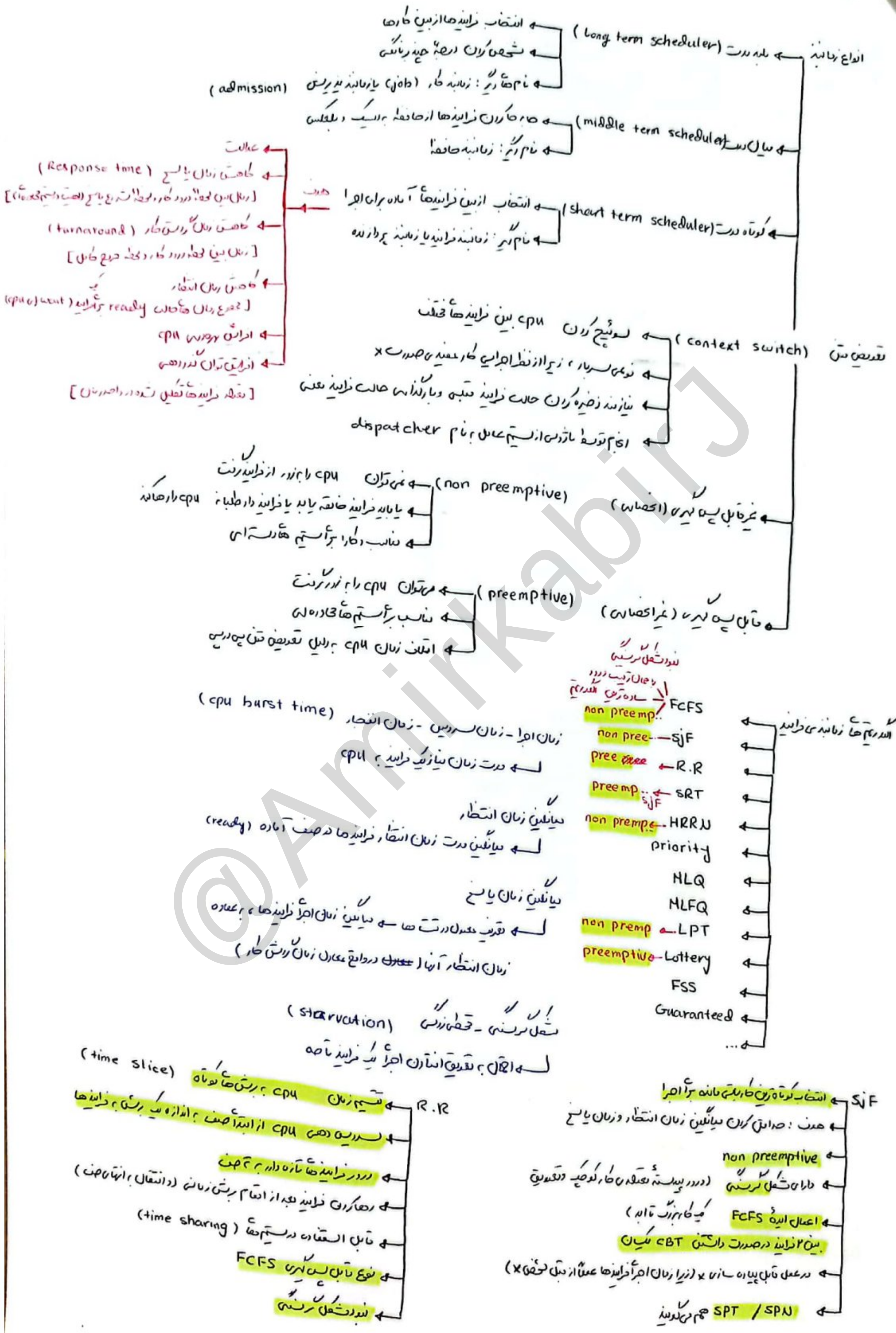
- 8 رخ دادن رخدادی که فرایند منتظر آن است
- 9 هیچ فرایندی در حالت ready نیست
- اولویت فرایند و ready از فرایند های ready با ترتیب

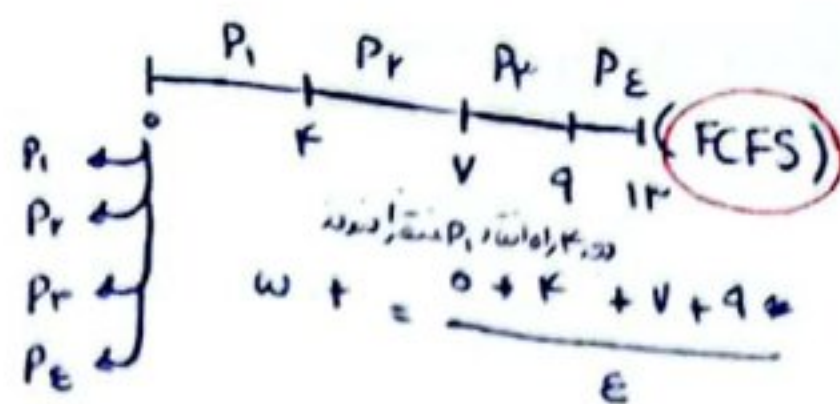
- 1 هر فرایندی که در حال انتظار است (یعنی فرایند با اولویت) به فرایند از صف انتظار انتظار شده و به یک منتقل شود (تا وقتی بر روی حافظه ایجاد شود)
- 2 فرایند از حالت صبر به ready و ready (در)

حالتی درون حافظه

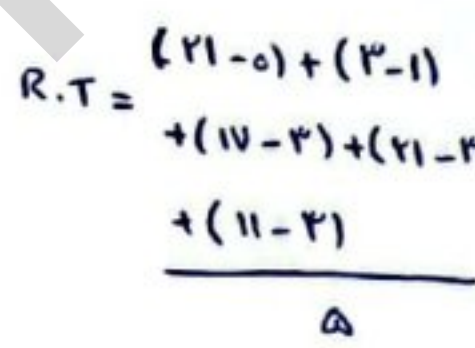
اولویت فرایند

فرایند در حال اجرا و منتظر است به سبب نیاز دارد از نظر هیچ فرایندی در ready





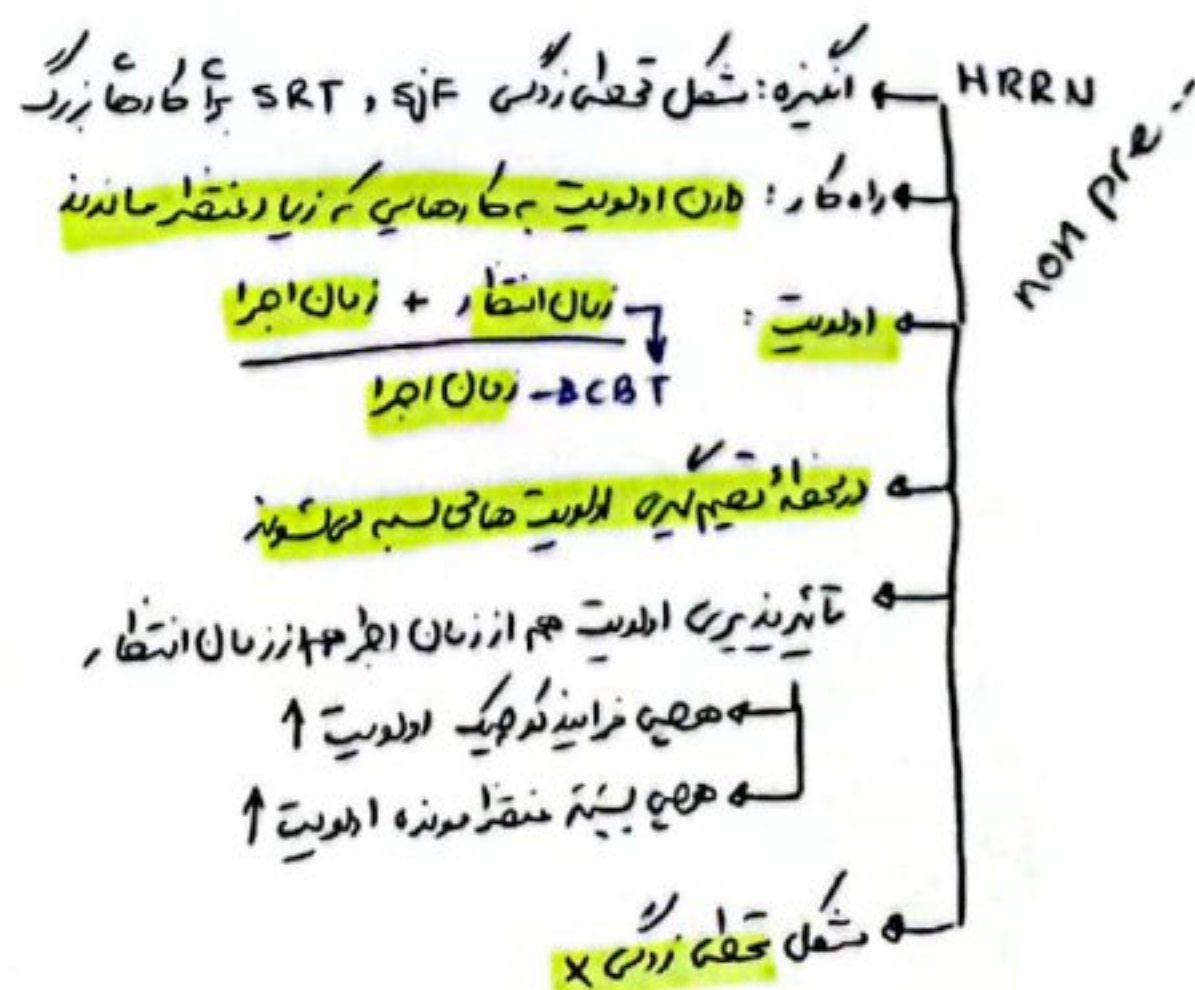
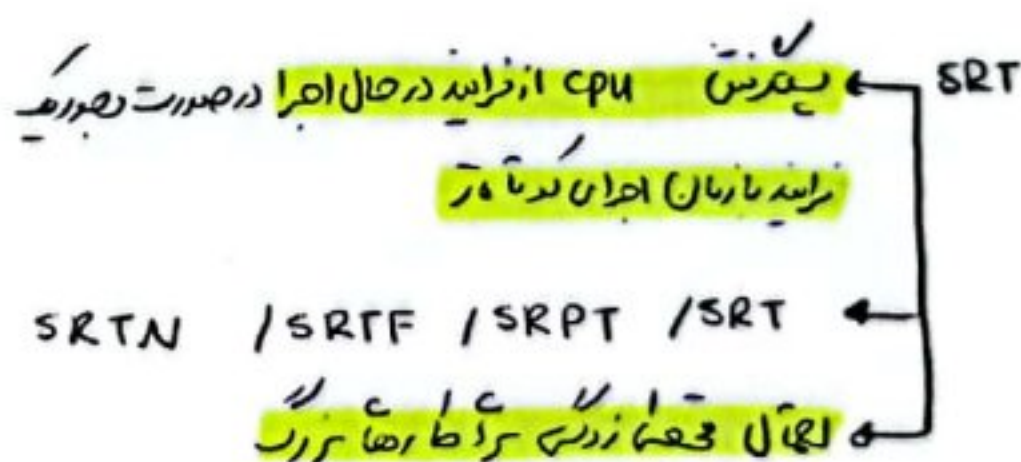
$1 \text{ min} = 60 \text{ s} \rightarrow 4 \dots \text{ms}$

[illegible]

وقت رست 1 ms ^۷ بیاض زرد یا سرخ ؟

R.R

فاز	CBT (min)	دور (min)
P ₁	۸	۰
P _۲	۱	۱
P _۳	۴	۳
P _E	۶	۳
P _Q	۲	۱۰



LPT
non
انتخاب طره نی ترین کار
اجال تحفه زدگی بر آکارها کوچه
مدان کردن میانین زمان یا نسخ زمان انتقا

MLQ
درسته بینه درایه ها در حبه صفت
دارن اولیت خاص در صفت
استفاده از یک الگوریتم خاص در صفت
تخصیص CPU بین صفت ها
non preemptive
استفاده از یک اولیت خاص در تحفه
preemptive
بیم R.R مثل ۵:۱ سهم صفت ۱
۳:۲ صفت ۲ ۲:۱ صفت ۳

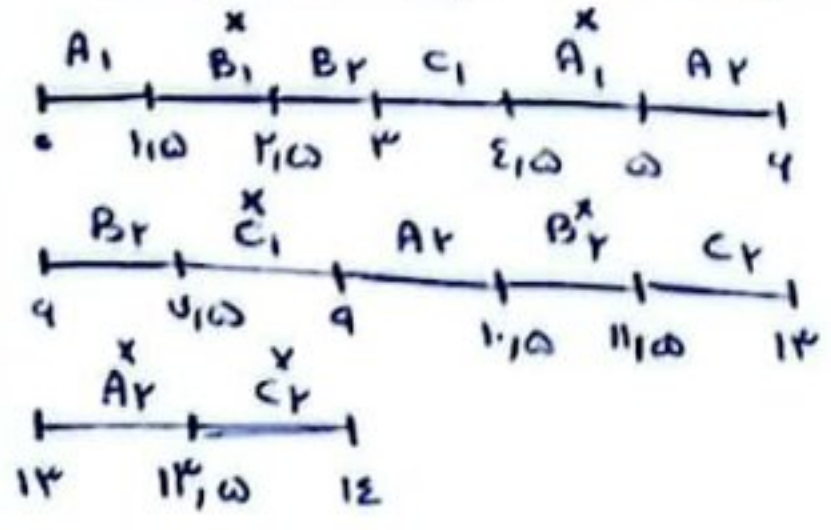
Lottery
non pre-
توزیع نقدیه بین خرابیه ها
بر اساس اولیت در میان نقدیه بلطف ها و خرابیه را تحفه
اعدام تعدادی یکس از بلطف ها به عنوان برنده
درایه های هر برنده زمانی (شماره ترافیک) بلطف برده اند
عدم وجود تحفه زدگی
امکان داره بلطف ها بین خرابیه ها
(مثلا خرابیه منتظر ۱۱۵ است بلطف ضرر اهدا)

MLFQ
اصلاح ایده MLQ با امکان خاص در صفت ها
لحم عنوان مثال یک خرابیه در صفت ها یا بین با آخرین زمان انتقا
بر صفت ها به تر منتقل می شود یا بالک
مثال: برترین سهم با ۴ صفت را در نظر بگرفت در صفت اول از R.R
برش زمانی ۴ms در صفت دوم از R.R با برش ۸ms در صفت سوم R.R
۱۴ms انتقا کرد در صفت اول به تر از سهم FCFS استفاده کرد حال
خرابیه ها به صفت ۱ وارد می شود اگر در این صفت به پایان رسیدند به صفت دوم
منتقل می شوند و اگر در این صفت به تر تکمیل نکردند به صفت سوم و چهارم وارد

FSS
تلاش بر آ رعایت عدالت
در نظر گرفتن ملک خرابیه بر سهم بلطف
(ایده صفت به ملک را در نظر خ)
اگرچه ۲ کاربر A و B ۴ کاربر A و B ۸ خرابیه B تحفه اخرابیه
دارد آیا باید ملک را در نظر بگیریم؟
مثال: اگر کاربر A ۴ خرابیه ۱a, 2a, 3a, 4a و B ۲ خرابیه b
اگر سهم هر کاربر ۵۰٪ باشد
a1, b1, a2, b2, a3, b3, a4, b4, ...
اگر سهم کاربر A, B ۷۵٪ و ۲۵٪
a1, a2, a3, a4, b1, a5, a6, b2, a7, a8, b3, a9, a10, b4, ...

thread
در سطح کاربر
ایجاد انتقا - اعزام - زمانیه ها در صفت کاربر
(بدون اطلاع ها) در داخل یک خرابیه صورت می گیرد
همه نقطه خرابیه را می تواند
هم عنوان مثال اگر یک خرابیه در صفت اول به تر شده
و اگر آن OS در آن خرابیه (شماره ها) به تر می شود
در سطح هسته
طیفه اعمال در برید و زمانیه ها در سطح OS به تر می شود
همه CPU به تر می شود به تر می شود به تر می شود
اگرچه در سطح سیستم به تر می شود به تر می شود به تر می شود

thread
اخر اینها تحفه در یک صفت خاصه بیان از یک خرابیه
نیاز به حمایت سهم عادل جهت وجود حبه تحفه در یک خرابیه
درسته به تحفه ها به صفت خاصه بیان، تحفه ها به تر
فایل ها به تر
اختصاصی یک سیستم به تر به تر (P.C.I) و شبیه ها به تر
بر صفت
نام ها به تر: خرابیه به تر درن - ع - سهم به تر به تر
به تر - رسته - رسته - به تر به تر
مثال مثال: به تر به تر به تر به تر به تر به تر به تر به تر



۱۳ - ۲,۵ - ۵ = ۱
اگر زمانیه ها در صفت اول به تر می شود
در صفت اول به تر می شود و در صفت دوم به تر می شود
در صفت اول به تر می شود و در صفت دوم به تر می شود

زمانیه ها در سطح کاربر
مثال: خرابیه A, B, C در صفت اول به تر می شود
در صفت اول به تر می شود و در صفت دوم به تر می شود
در صفت اول به تر می شود و در صفت دوم به تر می شود

Process	CBT	سم	انتقا	زمانیه
A1	۲	۵	۱۳	۱۰
A2	۳	۱۳	۱۳	۱۰
B1	۱	۷	۱۱	۱۰
B2	۳	۱۱	۱۱	۱۰
C1	۳	۹	۹	۱۰
C2	۲	۱۱	۱۱	۱۰

۳ = CBT
A1, A2, B1, B2, C1, C2
B1 -> 3, B2 -> 1
C1 -> 2, C2 -> 3
۱۳,۵ - ۱,۵ - ۳,۵ = ۸,۵

زمانه‌های τ و τ_{cpu}

- در سیم‌ها non preemptive یعنی زمانه‌ها با طول بیند عموماً غیر ممکن است
- این که در سیم در صورت امکان، کارهای در سیم‌ها چند cpu می‌تواند به چند زمانه‌ها با طول مختلف سیم
- روان کار، هرگاه یک cpu آزاد است از این کارها باقی مانده طریقه‌ها را می‌تواند انتخاب کند
- این که در سیم انتظار طول زمانه‌ها می‌تواند به چندین وقت عمل کند اما زمان یا سیم یا ترتیب در ده

- هدف: یکده کارها را در طول زمانه‌ها (زمان در دسترس برای اجرا کارها)
- زمان بندی non preempt برای کارها و سیم (مثل LPT)
- زمان بندی preemptive برای کارها و سیم (مثل Bin Packing)

- توزیع کارها بین cpu ها به گونه‌ای که همیشه cpu ها کار کنند
- حداقل ممکن زمان طول زمانه‌ها
- نقشه: در سیم‌ها یک زمانه‌ها باید به سیم‌ها اجرا کند
- استاد باید بهترین طول زمانه‌ها را انتخاب کند

$$W_{opt} = \max \left\{ \frac{1}{H} \sum T_i, \max(T_i) \right\}$$

زمان اجرا کار، T_i

- انتخاب کارها و انتخاب cpu ها و ترتیب x_{ij}
- یکده آخر کارها برای یک cpu انتخاب می‌کند در صورت W_{opt} ها
- باقی مانده کارها cpu ها می‌تواند به سیم‌ها

- RPT در اصل برای سیم زمانه‌ها LPT است
- سیمی و سیم‌ها
- در این سیم‌ها، کارها را در cpu ها می‌تواند به سیم‌ها
- به سیم‌ها در سیم‌ها
- در صورت انتخاب یک cpu است LPT در سیم‌ها

نشان: LPT با bin packing

$P_1=12, P_2=10, P_3=8, P_4=4$ cpu_1, cpu_2
 $P_5=3, P_6=3$

حداقل cpu ها

$$W_{opt} = \max \left\{ \frac{1}{2} (12+10+8+4+3+3), \max \{12, 10, 8, 4, 3, 3\} \right\} = \max \{21, 12\} = 21$$

$$W_{opt} = \max \left\{ \frac{1}{4} (24), \max \{12, 10, 8, 4, 3, 3\} \right\} = \max \{6, 12\} = 12$$

$P_1(12)$	$P_2(10)$
$P_3(8)$	$P_4(4)$
$P_5(3)$	$P_6(3)$

ω	γ	γ
κ	γ	

LPT 9

γ	γ	ω
γ	κ	

SPT 9

γ	γ	ω
γ	κ	

ω	γ	κ
ω	γ	
ϵ	ϵ	γ

γ	γ	ω
γ	κ	

ω	γ	γ
κ	γ	

ω	ϵ
ω	ϵ
γ	γ

$P_1(12)$	$P_2(10)$
$P_3(8)$	$P_4(4)$
$P_5(3)$	$P_6(3)$

$$W_{opt} = \max \{1, 5, 12\} = 12$$

$P_1(12)$		
$P_2(10)$	$P_3(8)$	
$P_4(4)$	$P_5(3)$	
$P_6(3)$	$P_7(3)$	

نشان: LPT همیشه زمانه‌ها با طول $\{12, 10, 8, 4, 3, 3\}$ را در سیم‌ها

نقشه: LPT و RPT با سیم‌ها زمانه‌ها

یکده به سیم‌ها و زمانه‌ها با سیم‌ها LPT و سیم‌ها
 (Reverse Processing time) cpu ها

یکده کارها و سیم‌ها زمانه‌ها RPT

از این سیم‌ها زمانه‌ها و سیم‌ها

- (*) اول به سیم‌ها cpu_1 و cpu_2 در سیم‌ها
- آخر LPT است به سیم‌ها LPT در سیم‌ها
- به سیم‌ها LPT در سیم‌ها

۱۰۵. OS سے سمافور سے برائے کم ساز، اجزاء (یعنی ترتیب خاص برائے اجزاء)

بانی

لے سٹا رہے

تاریخ کشته صف کشته - از این صراط ، دو کلام در یک صفت اعراس کشته

۵- اِزْرَائِيْلُ يَلْمِ اَصْلَاحَاتِ تَوْبَةٍ فِي كَلِمَةِ دِيْنِ اِزْرَائِيْلِ مُصَوِّفِي كَلِمَةِ

فمنها ما هو من أصول العقائد

فلم يزل ينادي يا رب اني ماضون في القفار

فرزندها را با نر است و آنرا ماضول النصاره
فرزیده طاعتی تولید کند به طایفه ماندر چنانچه آن ماندر ارضاء می کند

له وقتے مانر پر اس فرامیہ تولیہ لکھہ اور یہ

فرايند معرفت گشته، به شرط خاصي بعد از مائري كه تمام اصناف سه از مائري برسي دارد

لے دستہ حاضر خانیہ است مصروف کشته نمی تواند کارها کند مامور لید کشته شد

وہابیہ کا یہ اسماں ہے

۳۔ سمارٹ پول = Full
n = سمارٹ پول = empty

موتور = motor
M.F (موتور)

read count = 0
read count = 1

HE

فوائد - نویسنده : حیدر علی خان صاحب ضلعه و دیوان مرآتیه باب اصحابه و انجوائه

۱۰۰ فرضان نقطه در ضلع AB

۴۰۰ هزار ریال در اصل نقد بانک (اصلی است) باب ۱۰۰۰ ریال به ریال ریخته به ریال ریخته، نباید ریختن در آن

См. рис. 1

reader لا يقرأ

۴۔ اردو writer

لے بدن ادبیت

آریک ضابطہ اور یہ: اہندیہ اراکس، اہندیہ انتظار

البرهان الأول: ان الله تعالى قد خلق كل شيء بحدود وكلمات

۱- رتبه بیا، آراسته بیا، ستوده بینه

← يَمْنَنُ لِي صَدَقَ مَا أَتَقَا،

وَأَنَّكُمْ رَايَهُ نَسْتَعِظُكُمْ طَع

فیلو سائنس خزانہ سے حاصل کردہ رقم

40

از نظر مدیریت حافظه OS سه مدیریت حافظه اصلی

۱- حافظه ای (CPU) برادری

۲- دستورالعمل ها و داده ها سیستم عامل و برنامه ها

(رمز RAM)

↓

↑ سرعت

ثبات

حافظه رم

حافظه اصلی

حافظه مداخل

ایستادگی

دیده نوری

نوار مغناطیسی

برخی از مسائل مطرح در مدیریت حافظه اصلی

۱- در آن راه چینه برنامه در حافظه مانده ؟

۲- سهم بدی برنامه ها در آن از حافظه ؟

۳- کدام قسمت ها در اختیار کدام برنامه ؟

۴- مدیریت بخش ها حافظه

۵- کنترل در در واقع برنامه ها و از حافظه ...

آدرس ها در برنامه نوشته شده توسط برنامه نویسی به صورت عناوین هستند به $int i$ در این حالت این نام عناوین برای بهایی خاص از حافظه است.

در آن که کار $\rightarrow$ برنامه نویسی آدرس ها عناوین استفاده می کند

$\rightarrow$ کامپایلر آدرس ها عناوین را به آدرس ها قابل جا جایی تبدیل

$\rightarrow$ لودر باید آدرس ها قابل جا جایی را (مستطام اجرا) آدرس ها مطلق تبدیل

تبدیل تبدیل بنام آدرس به آدرس مطلق $\rightarrow$ زمان کامپایل

۱- اگر در زمان اجرا ترجمه تحفی باشد به برنامه در چه بخشی از حافظه قرار می گیرد

در این صورت که مطلق تولید می شود

این برنامه ها همیشه باید در یک جا ثابت بارگذاری شوند

۲- برنامه هایی از نوع COM در سیستم عامل DOS از این نوع هستند

زمان بارگذاری

۱- اصطلاح ترین حالت

۲- کامپایلر به قابل حمل آدرس ها بنام تولید می کند

۳- برنامه ها EXE در DOS-OS از این نوعند

۴- زمان اجرا $\rightarrow$ حالت بهای خاص، زمانه برنامه در حین اجرا معانی را تغییر دهد

حافظه
OS
برنامه کاربردی
بایستاده

۱- بخش مدیریت حافظه OS

۲- در اختیار برنامه کاربردی

۳- مانع از استفاده

کنش کابیزانت (overlay)

۱- هدف: اگر برنامه ها بزرگ تر از حافظه

۲- راهکار: هر زمان داده یا کد از برنامه لازم است

دارد حافظه شود در واقع ما جایی حافظه نداریم که فرایند به حافظه منتقل می شود

اما هنگامی که بخش کد نیاز باشد آن بخش به حافظه می آید

* یک برنامه ط حافظه مدیریت OS را می تواند در بر بگیرد

* عدم استفاده بهینه از حافظه

* محدود بودن اندازه برنامه به اندازه حافظه

میتواند اندازه‌های

حضرت نذاریں مازم : دیکھ

فراسد اعازہ ابراہیم

مجلس شورای اسلامی

مذہب دین و ایمان حافظہ مسئول

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

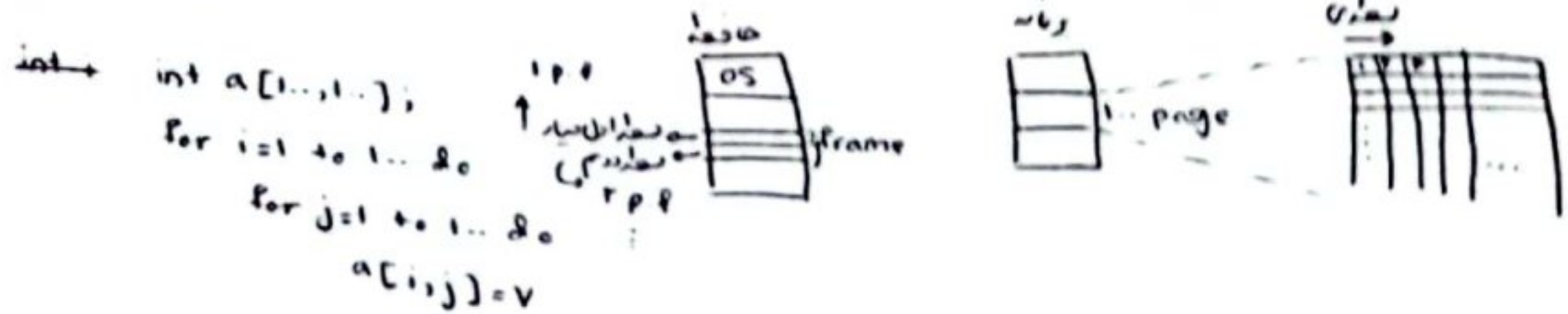
1

2

در حالت اصول ساده برنامه نویسی می توانیم در **آبرفت** برای **page fault** تاثیر بدهیم

اگر $array[0..100]$ داریم که هر عنصر ۲ بایت و اندازه هر صفحه ۲۰ بایت است. اگر فرض کنیم در صفحه ۱

فرض کنیم که می توانیم آرایه را به صورت صفحات در حافظه. یعنی فقط ۲ frame برای داده ها داریم



اگر $a[i,j]=v \rightarrow$ page fault ۱۹۸

اگر $a[j,i]=v \rightarrow$ page fault ۱۰۰۰۰

این تفاوت به دلیل این است که در $a[i,j]$ برای هر i فقط یک بار j را می بینیم و در $a[j,i]$ برای هر j ۱۰۰ بار i را می بینیم. این یعنی در $a[j,i]$ برای هر j ۱۰۰ بار i را می بینیم و در $a[i,j]$ برای هر i فقط یک بار j را می بینیم.

در سیستم از صفحات به ترتیب در هر صفحه می بینیم

در **FIFO** (First In First Out) سیستم، در هر **page fault**، اگر در **frame** جای **page** خالی باشد، آن **page** را جایگزین می کنیم. اگر **frame** پر باشد، **page** که از قبل در آن بوده است را حذف می کنیم و **page** جدید را جایگزین می کنیم.

OS	۴	۳	۲	۱	۴	۳	۵	۴	۲	۲	۱	۵
Frame												
#1	۴	۴	۴	۱	۱	۱	۵	۵	۵	۵	۵	۵
#2		۳	۳	۳	۴	۴	۴	۴	۴	۲	۲	۲
#3				۲	۲	۲	۳	۳	۳	۳	۳	۳
Page Fault	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×

page fault ۱۹۸

Belady Anomaly

OS	۳	۲	۱	۴	۳	۱	۲	۵	۴	۴	۱
#1	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳
#2		۲	۲	۲	۲	۲	۲	۲	۲	۲	۲
#3			۱	۱	۱	۱	۱	۱	۱	۱	۱
Page Fault	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×

page fault ۱۹۸

Belady Anomaly

در سیستم **optimal** (بهترین سیستم)، در هر **page fault**، **page** که در آینده کمترین بار استفاده خواهد داشت را حذف می کنیم و **page** جدید را جایگزین می کنیم.

OS	۳	۲	۱	۴	۳	۱	۲	۵	۴	۴	۱
#1	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳
#2		۲	۲	۲	۲	۲	۲	۲	۲	۲	۲
#3			۱	۱	۱	۱	۱	۱	۱	۱	۱
Page Fault	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×

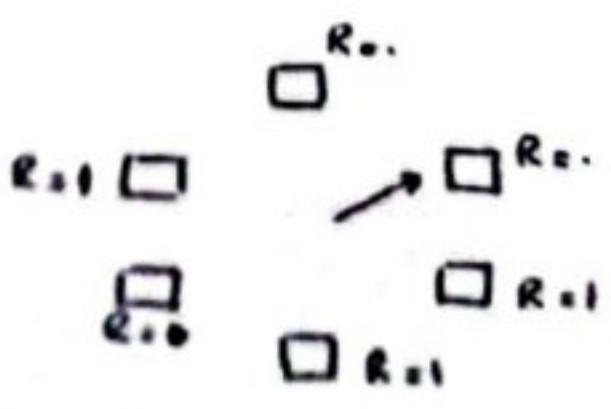
optimal

LRU

Belady Anomaly

OS	۳	۲	۱	۴	۳	۱	۲	۵	۴	۴	۱
#1	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳	۳
#2		۲	۲	۲	۲	۲	۲	۲	۲	۲	۲
#3			۱	۱	۱	۱	۱	۱	۱	۱	۱
Page Fault	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×

Frame
لیست



Second chance

در سیستم **Second chance**، در هر **page fault**، **page** که در آینده کمترین بار استفاده خواهد داشت را حذف می کنیم و **page** جدید را جایگزین می کنیم. این سیستم شبیه به **LRU** است، اما با یک تفاوت: اگر **page** در **frame** وجود داشته باشد، اما در **لیست** نباشد، آن **page** را به **لیست** اضافه می کنیم و دوباره بررسی می کنیم.

Second chance

second chance

✓ belady w. p. i. 4

اگر صفحه‌ها را با دار صاع این س‌ن را دار کم در صافه باقی بماند پس از $R=0$ و تا زمانی که دو بار باقی‌مانده صاع
اگر $R=1$ در صاع کم

العجز
(aging)

[illegible]

بعد از هر کشف $R=0$

مهرتوان گفت این لحظه ها مآخیزه ارجاعات دانشه داری

os صفحه fault صفحه ای را انتخاب، شماره ذکر دارد

سَمَاعُودُهُ لَوْحَتُهُ فِيهِ اَصْحَابُ كَمَرِ السَّعَادَةِ لَهُ

در این صفحه اسم شما را بنویسید آن ۱۰۰۰ ۰۰۰۰ اس (اولیاد) بنویسید

مادون در صافه ثبت به شماره ۵۱۱۱۱۱۱
به این مال لغیر

هنگامی که سازه در حالت پایداری قرار دارد، f_{iP} مقدار

این نوعی پیاده سازی برای LRU

احرفه از است و ضفت M و R

هر چه از آب و صفت ۴ و ۸

set M $\rightarrow$ $\{m_1, m_2, \dots, m_n\}$

۱۰ این صفتی، اضرار حاصله، باد

[illegible]

صفحه پیدا یا باید از بین اداسی، ارجاع به صفحه ای، تغییر شماره حزب (زیرا نوشتن دوباره همان روش disk باز)

✓ Belady's rule

کاربرد نسبی

گروه : اراغند، ها = اولین ادویه برآغند

کمره ۱ : عارضات در کمره
کمره ۲ : عارضات در کمره

۱- ضوابط ادعای کرده هم در این ادله است

۳۵۰: عذرتن افرا رصوع - چارسین ادولیت

و اگر فرض کنیم شماره OS صورت دره A است R مربوط به صفحه را به شماره آن اضافه و بیت $R = 0$ باشد

LFU / NFU

۴. نمایندہ نقولہ ارجاعات این صفحہ را بنمایش

۴ دھندلے، اسماء کتر، راضیہ انتصاب

لرځه پیاوړه لارې LRU

این الگوریتم در واقع چیزی را فراموش می‌کند $\rightarrow$ معنی در یک بازه حین آن بدکاران $=$ سنجیده $\uparrow$ اما در این الگوریتم در واقع $\rightarrow$ این الگوریتم از LRU

MFU

مثال LFU یک سیماینده برای مصدنه دار

Page-Pault برآ صند انتخاب نمایند کرد

اینکه این آیه، صدها سالها در نزد مردم و در اندازه طایفه در حلقه بوده و باید که این آیه از حلقه را به بقیه بدیم

M	R	on
e	o	o ²
o	i	i
i	o	r
i	i	r

دکمه نیاز به حافظه جانبی

حافظه اصلی حجم محدود

OS داده ها را در قالب فایل ها بر روی حافظه جانبی (disk) ذخیره

بخشی از OS، فایل ها را بر روی دیسک و سیستم فایل

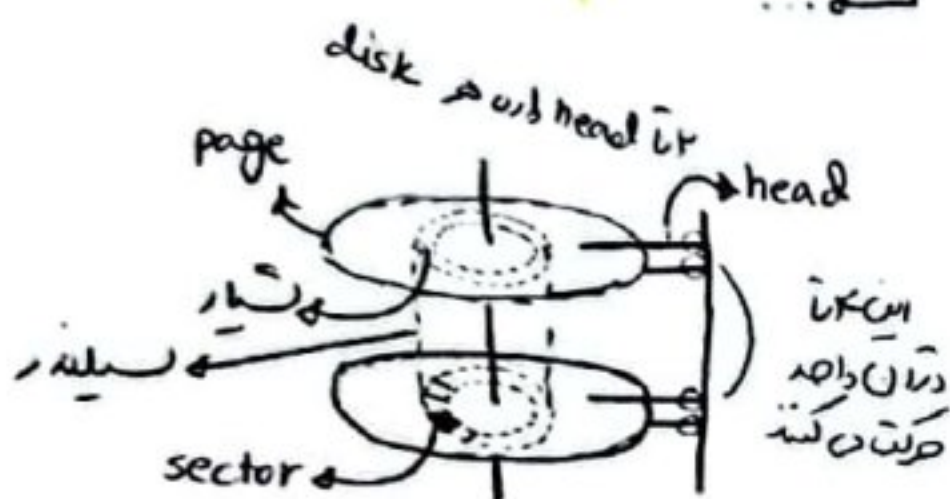
انواع حافظه جانبی

- نوار مغناطیسی
- دیسک مغناطیسی
- دیسک نوری
- حافظه فلش
- SSD

دسترسی ترتیبی زمان دسترسی به داده به همان آن بستگی

نوار مغناطیسی

سیستم (مصادره) RAM



دیسک مغناطیسی

یکه از رایج ترین روش ها ذخیره سازی در سیستم ها حافظه ترا (تیمت در سرعت و مقیاس)

از نقطه دید دور شکل

بر روی هر صفحه یک head ذخیره write, read (head می تواند روی چند حرکت)

هر صفحه به بارها که همانند یک نام یکبار تقسیم (بیردنن ترین نیاز، نیاز ه)

هر نیاز به سمت هاست sector (نقطه sector در هر نیاز هاست)

نیازها هم شکل بر روی صفحات مختلف شکل - سیلندر

دیسک 14 page نیاز هر صفحه 2.48

sector 42 اندازه 012 byte نیاز یک دیسک

$$14 \times 2.48 \times 42 = 14 \times 103.76 = 1452.64 \approx 1453 \text{ sectors}$$

14 page، هر سیلندر 4.94 sector

هر نیاز چند sector ؟

$$4.94 \times 14 = 69.16 \approx 69 \text{ sectors}$$

وقت زمان دسترسی به disk زمان جست و جوی (seek time) زمان دور نیاز برای انتقال باز روی هر نیاز (سیلندر) ترا هر write, read

زمان تاخیر چرخشی (rotation latency) زمان دور نیاز برای چرخش disk (تا هر در sector تکرار)

زمان انتقال زمان جابجایی اطلاعات (بستگی به سرعت چرخش disk)

الگوریتم های زمان بندی FCFS SSTF Scan

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

head در یک جهت حرکت در راه خود تمام درخواست ها را پاسخ تا جائیکه آخرین نیاز برسد به جهت تغییر حرکت 1) 10 - 92 - 82 - 51 - 23 - 1 - 7 - 5

Scan C-Scan look C-look N-step-Scan

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

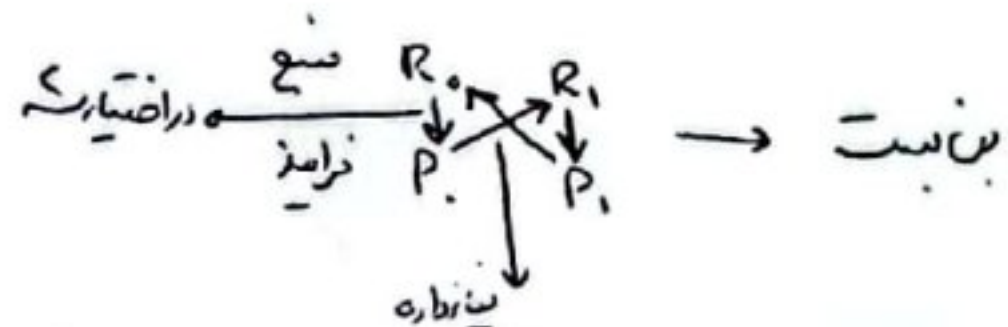
- تعریف فایل**
- یک یا چند ذخیره ساز منطقی، خصوصیات فیزیکی و سخت افزاری را از ذخیره ساز رایزن
 - جموعه ای از اطلاعات مرتبط، بررسی حافظه ماند ذخیره
 - نویسه های واحد تخصیص حافظه ماند؛ صورت منطقی
 - در فایل دنباله ای از بیت ها، معنا واحد دارند این معنا و مفهوم توسط زنده فایل مشخص

- صفات فایل**
- نام (معمولاً ۲ بخشی)
 - نوع
 - شماره
 - مکان ذخیره ساز
 - اندازه
 - کاربر (مالک، ایجاد کننده)
 - مولفه ساز زمان مکان
 - اطلاعات کنترل دسترسی و حفاظت
- عملیات فایل**
- ایجاد (تخصیص معنا)
 - read
 - write
 - حذف
 - delete
 - حذف کردن ...

- باختارهای فایل**
- باختار با توانایی
 - " " رکورد
 - باختار دفته ...
- انواع دایرکتوری**
- ساده (خفا یا نه)
 - خفا (نه درختی یا درختی)
 - گراف بدون دور
 - گراف عمیق
- مفهوم دایرکتوری (یا Folder) به راه کار بر اساس زنده فایل ها**

سیستم فایل به بخشی از OS که مدیریت می کند ارتباط با فایل برقرار

- وظایف سیستم فایل**
- زنده فایل دایرکتوری
 - مدیریت نحوه ذخیره سازی فایل ها در disk
 - مدیریت فضای آزاد disk
 - انتخاب فایل ها بین برنامه ها و کاربر
 - کنترل دسترسی و امنیت از فایل ها
 - غیریک ساختار مناسب برای کاربر ...



بن بست - خطی ها منتظر یکدیگر باشند

در صورتی که در بن بست، هر یک از فرایندها این مجموعه منتظر رود و این هستند که منتظر فرایند دیگری از همین مجموعه می توانند آن را بکار

(استفاده اشتراکی)

اکتسار متقابل
(Mutual Exclusion)
(M.E)

شرط
کافی
ضروری
نیروی
بن بست

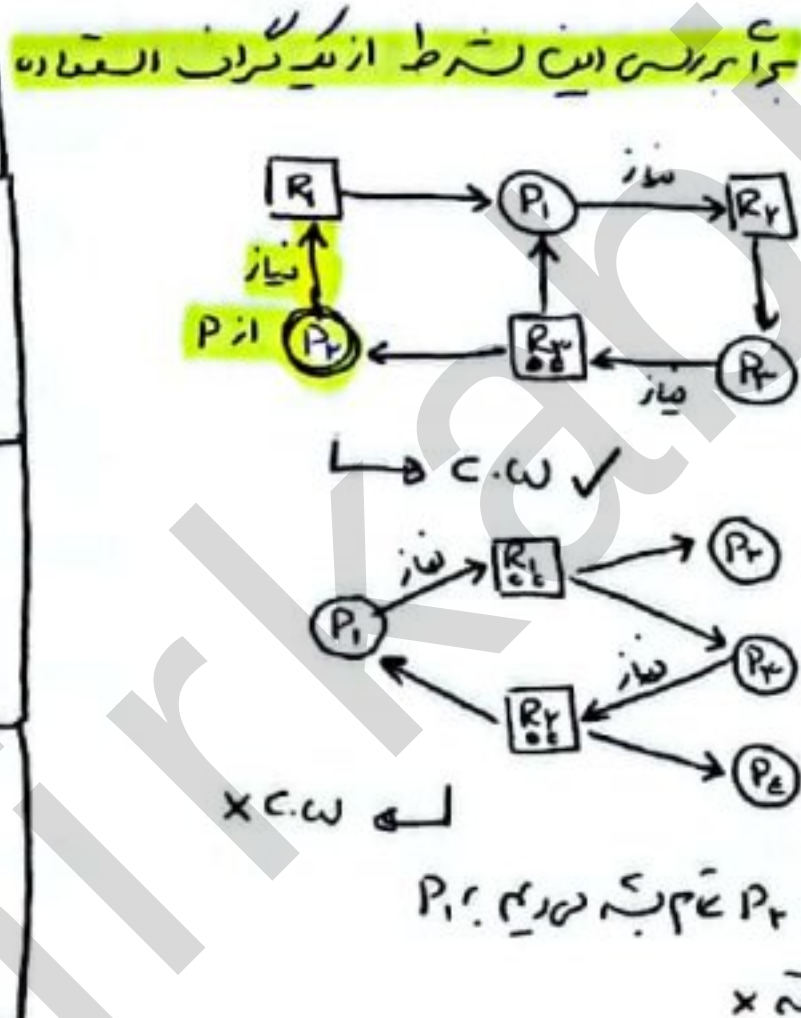
فرایند یک منبع را در اختیار دارد و می تواند به سایر منابع جدید نیز در اختیار دهد و به این ترتیب منابع دیگر را می تواند

غیر قابل پیش گرفتن
(non preemption)

در صورتی که منابع دیگر از آن

استقرار چرخشی
(Circular wait)
(C.C.W)

در صورتی که از فرایندها در اختیار دارند و هر یک از فرایندها در اختیار منابع دیگر از همان زنجیره است (حالت)



این گراف مجموعه ای از لبه ها و این ها است
رئوس این گراف
فرایندها، منابع
منابع هستند
منابع، منابع

لبه ها این گراف
از یک منبع به یک فرایند و به منابع دیگر اختصاص
در صورتی که در
از یک فرایند به یک منبع و به منابع دیگر اختصاص

اگر از یک منبع چندین در دسترس در داخل رگ های مربوط به آن منبع
از منبع نقطه گراف استفاده
شرط C.C.W
در صورتی که از هر منبع نقطه یک منبع باشد

روش ها برای پیشگیری از بن بست - الگوریتم کشه مرغ (ostrich) - در صورت وقوع بن بست سیستم دوباره از ابتدا شروع
در سیستم های بن بست به سرعت باید بتوان

کشف در سیستم بن بست
(Detection & Recovery)

پیشگیری از بن بست
(Deadlock Prevention)

اجتناب از بن بست
(Deadlock Avoidance)

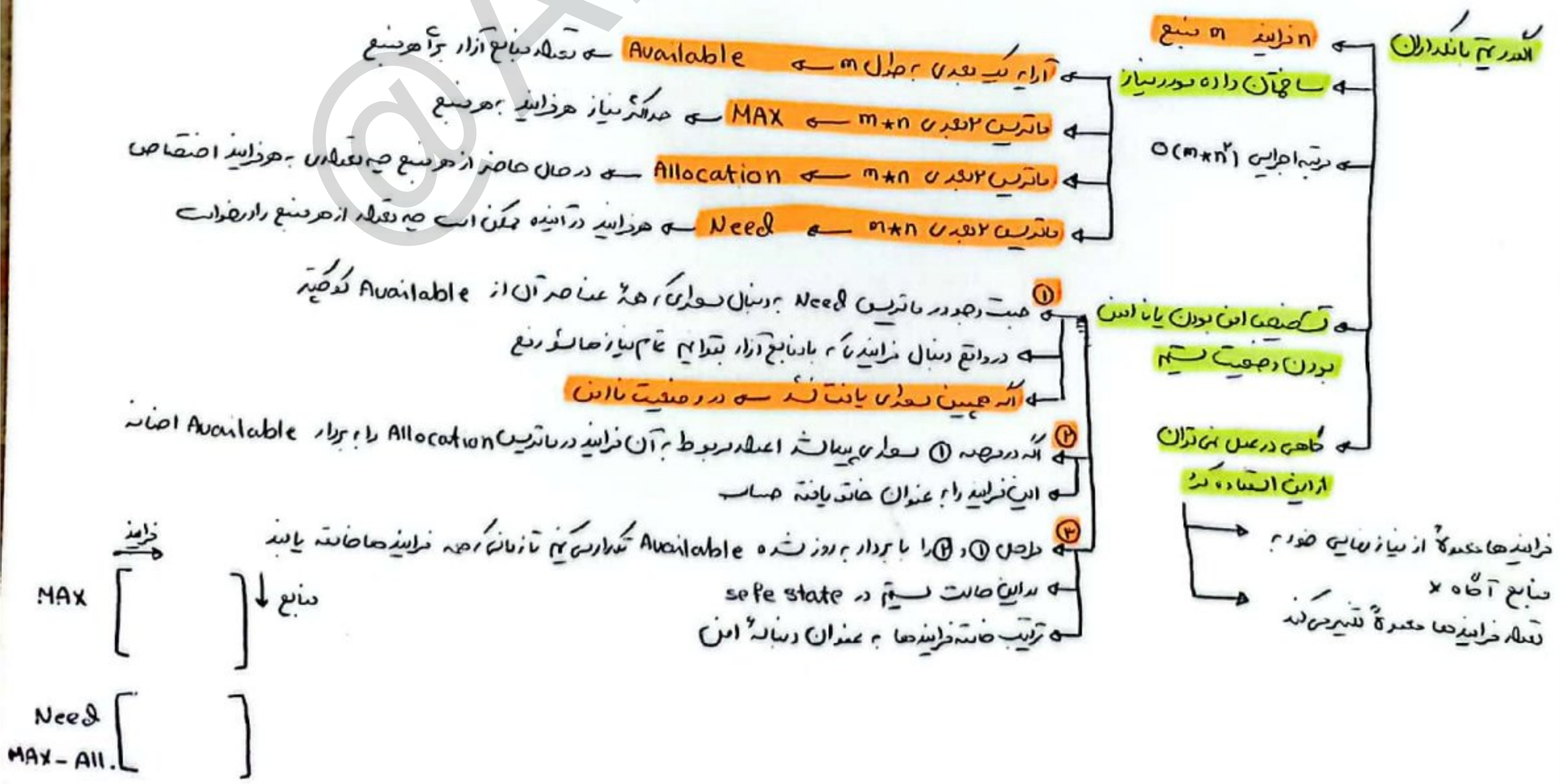
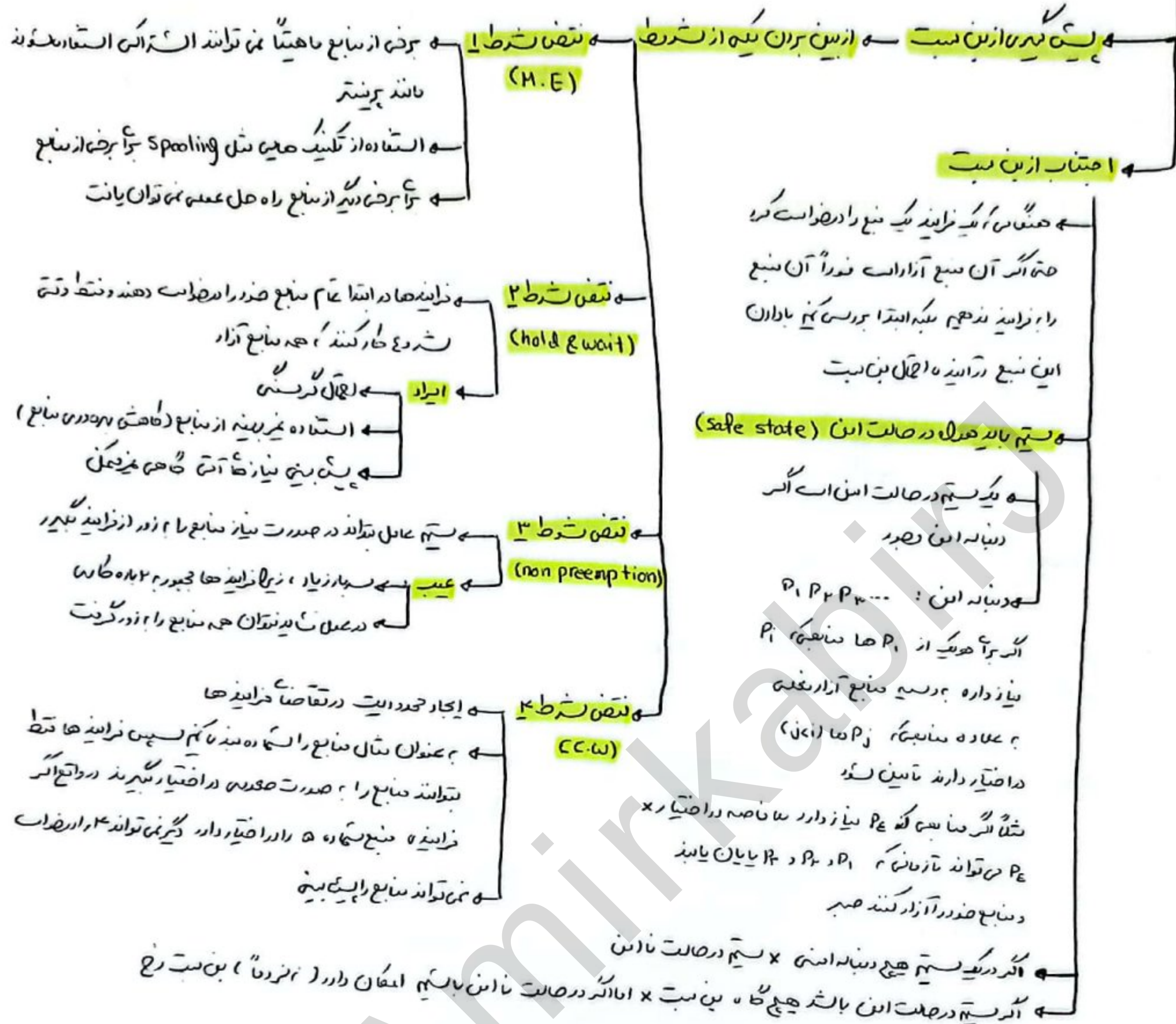
تخصیص بن بست
حالت اول: اگر منابع یک منبع به ما استفاده از گراف تخصیص منابع و در صورتی که

الگوریتم تخصیص بن بست
مشارکت یا رابطه های زیاد

منابع یک منبع در هر لحظه قابل اعطای اجراء
منابع، منابع
در بازه های زمانی تخصیص اجراء
در زمان تخصیص یا صحت و خطا
منابع، منابع

پیشگیری منابع
منابع را به زود از فرایندهای دیگر و فرایند
به عقب برگرد
تا در انتخاب فرایند ورودی کارایی سیستم

خانه می فرایند و از آن
منابع در اختیار آن
داده دوباره فرایند
حالتها
تا در نحوه انتخاب فرایند
بررسی کارایی سیستم



MAX

	A	B	C
P ₀	7	5	3
P ₁	3	2	2
P ₂	9	0	2
P ₃	2	2	2
P _E	4	3	3

Allocation

	A	B	C
P ₀	0	1	0
P ₁	2	0	0
P ₂	3	0	2
P ₃	2	1	1
P _E	0	0	2

Available

	A	B	C
(آزاد)	3	3	2

$P_1 \rightarrow P_E \rightarrow P_2 \rightarrow P_0 \rightarrow P_3$

بقیه در حالت امن است

MAX-All.

→ Need

	A	B	C
P ₀	7	4	3
P ₁	1	2	2
P ₂	4	0	0
P ₃	0	1	1
P _E	4	3	1

→ available

Available

A	B	C
2	1	2

Available

A	B	C
5	3	2

available

A	B	C
5	3	4

Allocation

	A	B	C
P ₀	0	1	0

Need

	A	B	C
P ₀	7	3	3

available

A	B	C
3	2	2

→ safe x

Allocation

	A	B	C
P ₀	0	3	0

Need

	A	B	C
P ₀	7	2	3

available

A	B	C
2	1	2

Need

available

A	B	C
5	3	2

$P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_0 \rightarrow P_E$

حالت امن است

✓ available

✓ safe state

حالت امن است

✓ available

x Need